Python for Data Analysis and Scientific Computing

What will we cover today

- About the course
  - About me, About you, your expectations
  - How this course has been put together
  - Rubric
- The Analysis Landscape
  - Tools for Analysis
  - Sample & Examples of various Analysis Efforts
- Practical Aspects
  - A Framework for any data analysis effort
  - Best Practices
  - Real Life Use Case

About me

- HELLO
  - My name is Pramod Gupta
- PhD in Electrical and Computer Engineering, from McMaster University, Canada
- Experience: Academics, NASA, EMC, GE, VISA, Startups etc.
- Teaching Data Science and Machine Learning Related Courses including R and Python
- Independent data science consultant

Philosophy

- Sharing what I have Learned
- Let us make this an interactive experience. I would like this class to be as open and interactive
- Feel free to stop and ask for clarifications
- If you have a doubt, always better to ask
- Don’t be shy or hesitant
- Knowledge flow is bi-direction. Age is never a factor in knowledge. It is the experience.
- Goal: “ Learning to self-Learn

Your turn

Quick Introductions

- **Name, Profession**
- **Experience in analysis do you have?**
- Which analytics tools have you used/familiar
- What are your expectation and motivation from this course?
- What are the areas where you would like to apply and analyze (you can upload to the class forum)

Course Logistics

Goals and objectives

- Data science is a huge field, and there is no way you can master it by reading a single book or taking a single course. The goal of this course is to give a solid foundation in the most important tools.

- This course is concerned with the nuts and bolts of manipulating, cleaning, and crunching data in Python. It is also a practical, modern introduction to scientific computing in Python, tailored for data-intensive applications.

- The goal is to offer guidance to the parts of the Python programming and its data-oriented library ecosystem and tools that will equip you to become an effective data scientist.

Goals and objectives

- This course is also about the parts of Python and libraries (Modules), you will need to effectively solve a broad set of data analysis problems.
- The class is designed to provide you with the tools you need for solving real world problems using statistics and a better understanding of data analysis techniques.
- We plan to achieve these goals by introducing you to the relevant statistical knowledge, how to use Python to perform data analysis and engage in solving problems, analysis through homework, discussion, class participation and project.

Goals and objectives

- Attain deeper understanding of the mathematical toolkit provided by the core packages in Python particularly NumPy, SciPy, Pandas, and Matlbaplotlib when dealing with heavy mathematical, engineering or scientific problems.
- Acquire in-depth hands-on experience
- You will learn mathematical operations with array data structures, pivoting, basic statistics, probability density functions, interpolation etc.

What will you learn

- First you must define your requirements or objectives. Once you have defined your requirements, next step is to think about the data. What is required and how to obtain it.

- Next you need to import your data into Python. This typically means that you take data stored in a file, database, or web API, and load it into a data frame in Python.

What will you learn

- Once you have imported your data, it is a good idea to tidy it. Tidying your data means storing it in a consistent form that matches the semantics of the dataset with the way it is stored. In brief, when your data is tidy, each column is a variable, and each row is an observation.

- Next step is to transform the data. Transformation includes narrowing in on observations of interest (like all houses in one zip code, all data from the last year), creating new variables that are functions of existing variables (like computing velocity form speed and time) know as derived variable, and calculating a set of summary statistics (like means or counts).

What will you learn

- Tidying and transformation are called wrangling, because getting your data in a form that is natural to work with often feels like a fight.

- Once you have prepared your data, with variables you need, there are two main engines of knowledge generation***: visualization*** and ***modeling***.

- **Visualization** is a very important. A good visualization will show you things that you did not expect, or raise new questions about data. A good visualization might also hint that you are asking the right or wrong questions, or you need to collect different data.

What will you learn

- The last step of data science **communication**, an absolutely critical part of any data analysis project. It does not matter how well your models and visualization have led you to understand the data unless you can also communicate your results to others. The format will depend upon the audience.

- Above all these tools is **programming**. Programing is a cross-cutting tool that you use in every part of the project. You do not need to be an expert programmer to be a data scientist, but learning more about programming pays off because becoming a better programmer allows you to automate common tasks, and solve new problems with greater ease.

- At the end of the course:
  - Be able to understand the concepts, strategies, and methodologies related to the design and construction of data mining/data analysis
  - You will be able to perform independent analysis of data
  - Be able to comprehend several data preprocessing methods
  - You will have the tools to tackle a wide variety of data science challenges, using the best parts of Python.

Grading

- Grading

– Homework	50%
– Project	50%

- Work Load
  - You are expected to put in 6-7 hrs. per week of work outside of class. A few of you will do well with less time than this, and a few of you will need more but do not worry.

Homework

- Homework
  - There will be 4-5 assignments and will comprise of problems related to the topic covered in the lecture. The objective of the assignments is to help you develop more in-depth understanding of the material covered in the lectures.
  - Do not copy someone else work, both parties will be penalized. Though you are welcome and encourage to work with each other.

About final project

- Aim: turning data analysis techniques you learn in class to become your strength in dealing real world problems

- The project involves analysis of the data, implementation, preparation of a report and presentation of the results during the last week of the class. The project will be done in groups of 2~3 students. If you already working on a research project in the area of interest you are encouraged to use dataset/topic from your research provided you make some extra effort for the class.

A data analysis write up

Should tell a full story

- Title, Introduction (Motivation)
- Method used (justification)
- Caveats
- Results
- References

Do not wait!
Get started early.
You will learn more that way

Who is this course for?

- Anyone who is interested in:
  - Helping companies make decisions aided by data
  - Refreshing some theory learned in school, but with a practical focus
  - Getting up to speed with new Open Source tools and libraries
  - Curious about the new technology

What kind of Data?

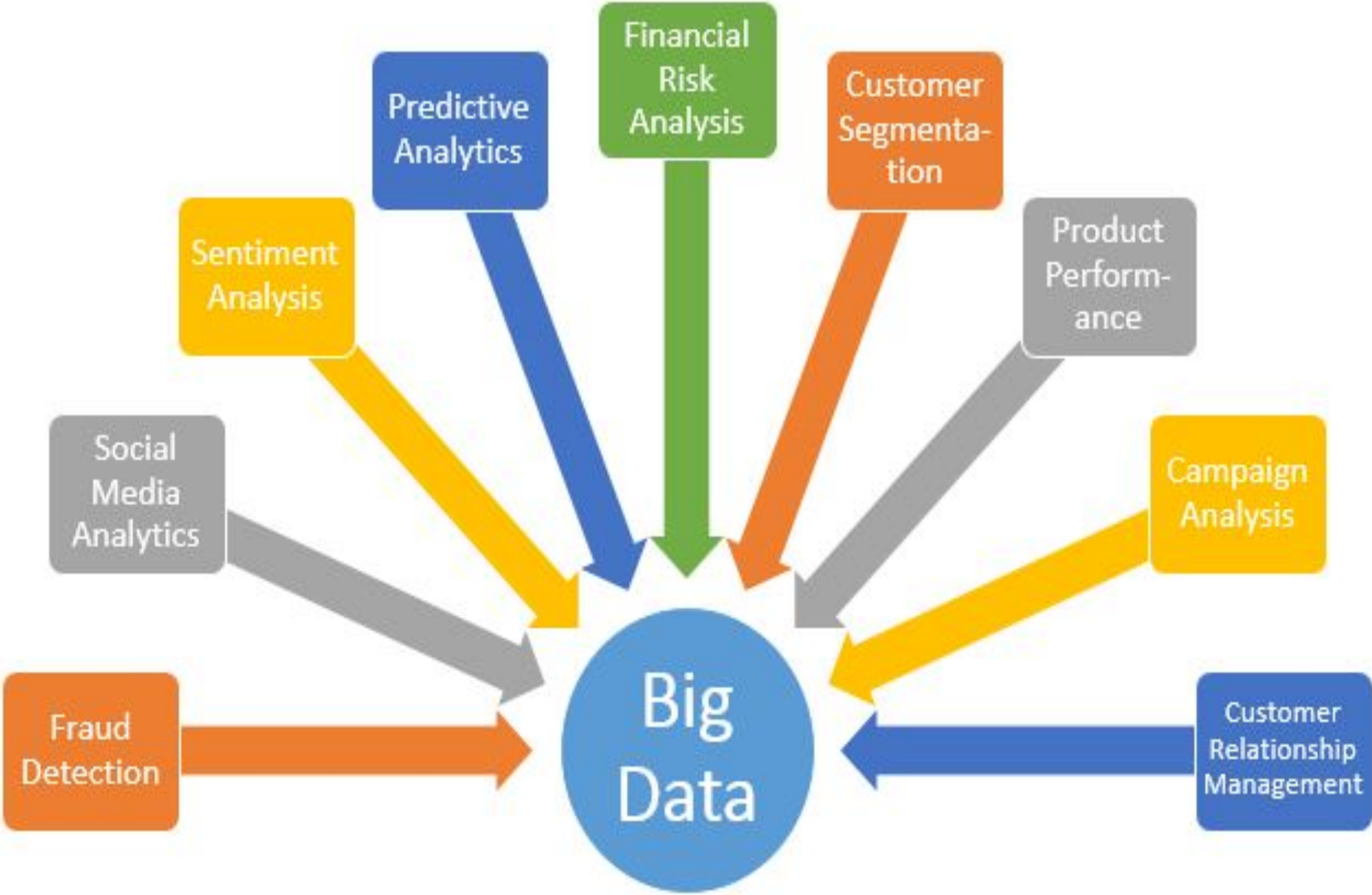
- When we say “ data”, what we are exactly referring?
  - It is the information which describes the object, e.g., car is described by color, mpg, gear, sedan, four door etc. The properties which describes the object are known as attributes/features/variables.
- The primary focus is on *structured data*
  - Tabular or spreadsheet-like data in which each column may a be different type (string, numeric, date, or otherwise). This includes most kind of data commonly stored in relational database or tab- or comma-delimited files
  - Multidimensional arrays (matrices)
  - Multiple tables of data interrelated by key columns (what would be primary or foreign keys for a SQL user)
  - Time series
- This is by no means a complete list. A large percentage of datasets can be transformed into a structured from that is more suitable for analysis and modeling.

What kind of Data?

- Sometimes it may not be possible to extract features from a dataset into structured form. E.g., a collection of news articles could be processed into a word frequency table, which could then be used to perform sentimental analysis. is by no means a complete list. A large percentage of datasets can be transformed into a structured from that is more suitable for analysis and modeling.

Data analysis is all around us …

- Typical areas where Data Analysis can be performed
  - Agriculture
  - Computing
  - Crime
  - Ecology
  - Health
  - Hydrology
  - Meteorology
  - Finance
  - Social networking



Financial Risk Analysis
Predictive Analytics
Customer Segmentation
Sentiment Analysis
Product Performance
Social Media Analytics
Campaign Analysis
Fraud Detection
Big Data
Customer Relationship Management

Software Choices For Data Analysis

Data Scientist’s Toolbox

Predictive Analytics Tools in Market
python
sas
MATLAB
WEKA
KNIME
rapidminer
IBM
SPSS
R
mahout
Minitab
SAP
StatSoft
STATISTICA

sas®

BI Development
Online Training

crystal reports®

+ a b | e a u
SOFTWARE

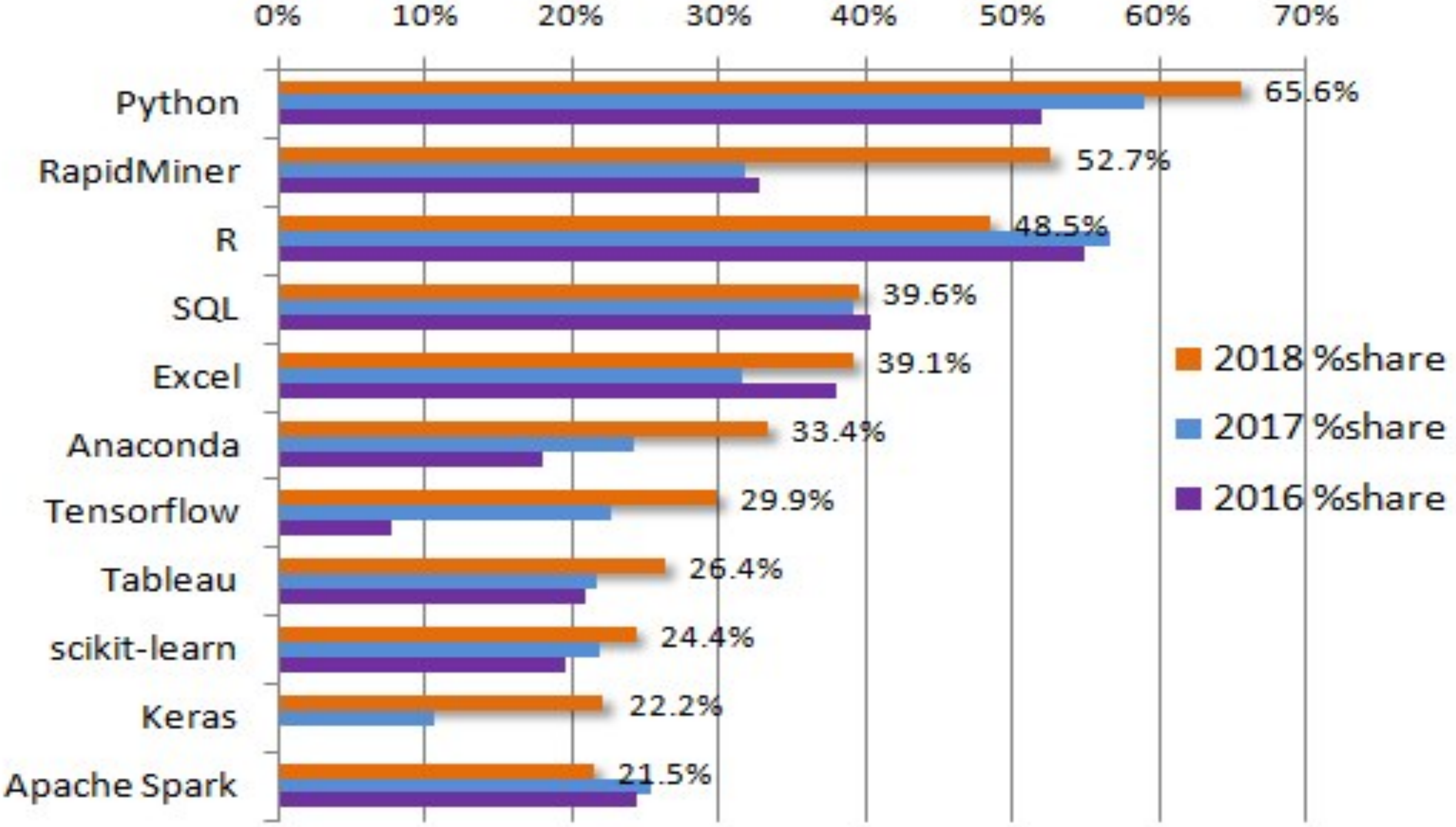
BI Tools

IBM
COGNOS

BRIO
SOFTWARE

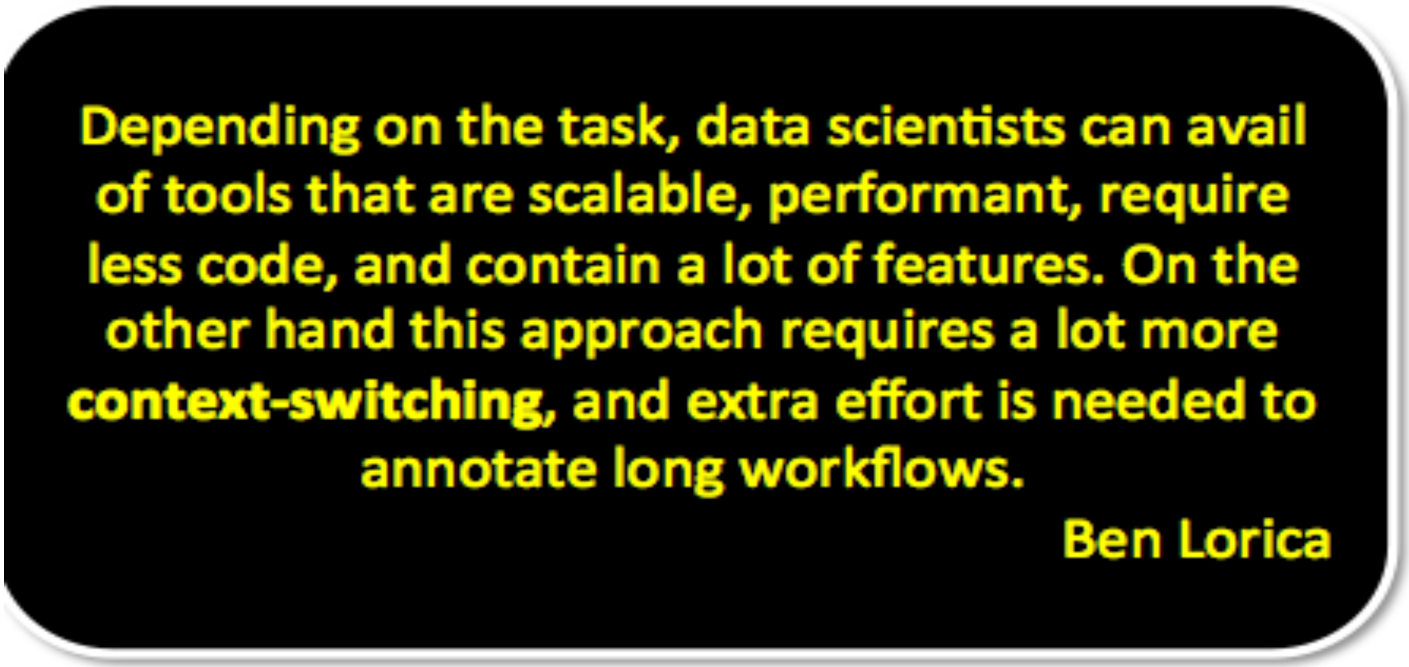
MicroStrategy®
Best In Business Intelligence™

KDnuggets Analytics, Data Science, Machine Learning Software Poll, 2016-2018



Focus on the results, not the tools

- It is okay to use multiple tools
- Whatever you know to get the job done
- Specialized Tools: The best tool for the right job



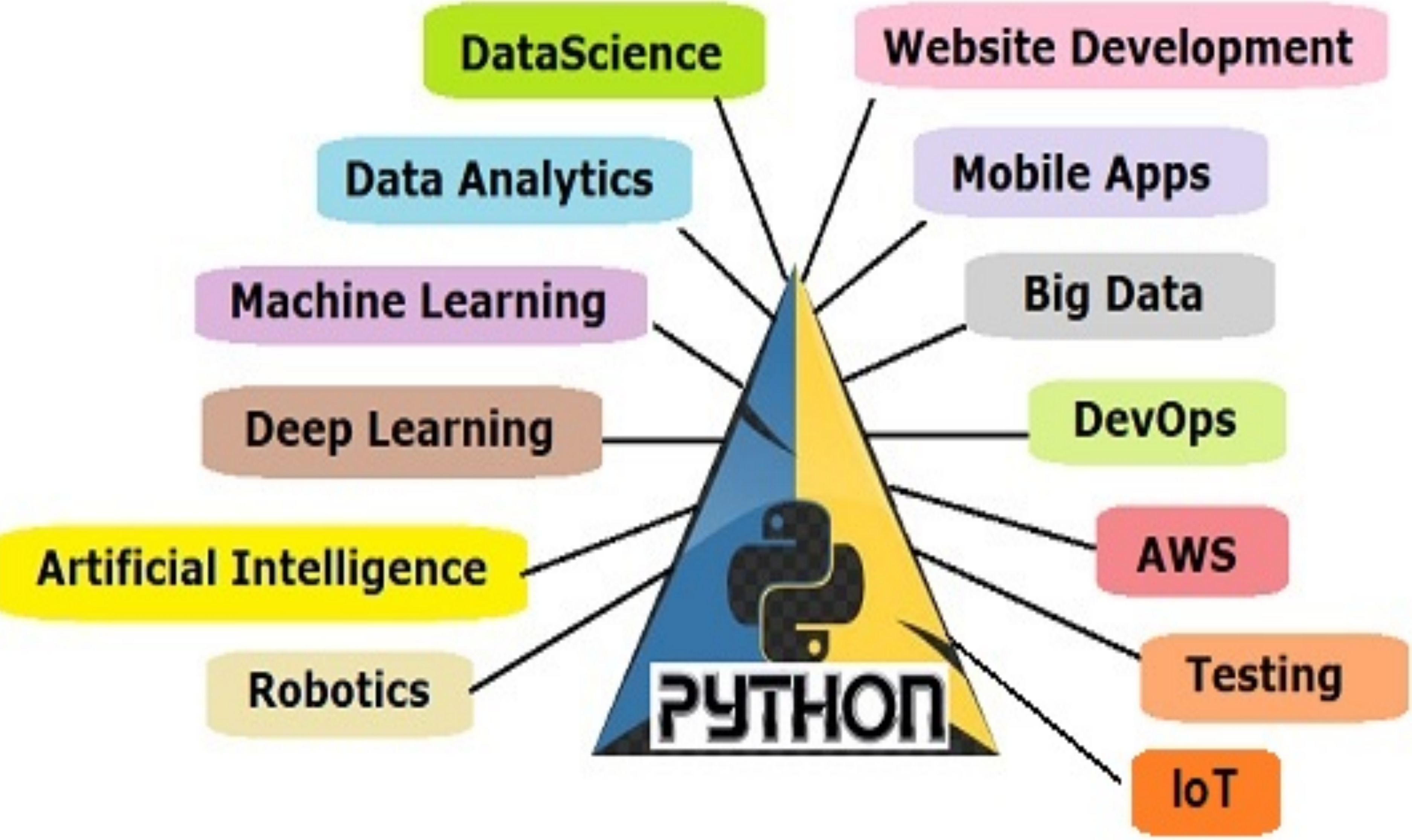
Depending on the task, data scientists can avail of tools that are scalable, performant, require less code, and contain a lot of features. On the other hand this approach requires a lot more **context-switching**, and extra effort is needed to annotate long workflows.

Ben Lorica

What is Python

- Some programming languages sit in the heart of data science. Python is one of those languages. It is an integral ingredient for Data Science and vice versa.
- Python is an object-oriented language created by Guido Rossum in 1989.
- It is ideally designed for rapid prototyping of complex applications.
- It has interfaces to many OS system calls and libraries and is extensible to C and C++.
- It is a very popular language as many companies use the Python (NASA, Google, Facebook, Linkedin etc.).
- Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields.

Cont'd


DataScience
Website Development
Data Analytics
Mobile Apps
Machine Learning
Big Data
Deep Learning
DevOps
Artificial Intelligence
AWS
Robotics
Testing
IoT
PYTHON

Why Python?

- Python provides great functionality to deal with mathematics, statistics and scientific functions. When it comes to data science application, it provides extensive libraries to deal with.
- It is not only open-source but also interpreted and high level tool
- Most importantly, Python is widely used in the scientific and research communities because of the ease of use, it’s simple syntax makes it easy to adapt for people who even do not have an engineering background.
- Thanks to huge user base, just about every function that you might need for data analysis is available, often through open source extensions (known as packages) made available by the community.

Why Python?

- In summary, some reasons which go in favor of learning Python
  - Open Source − Free to install and use
  - Awesome online community
  - Very easy to learn
  - Libraries and Frameworks - Python has numerous libraries for different needs. Django and Flask are two of the most popular for web development and NumPy and SciPy and Pandas are for Data Science.

Though there are many advantages, it has few drawbacks too:

  - It is an interpreted language rather than compiled language, hence might take up more CPU time. However, given the savings in programmer time (due to ease of learning), it might still be good choice.
  - Python can be a challenging language for building highly concurrent, multithreaded applications, particularly applications with many CPU-bound threads. The reason for this is that it has what is known as the global interpreter lock (GIL), that prevents the interpreter from executing more than one Python instruction at a time.

Why to use Python

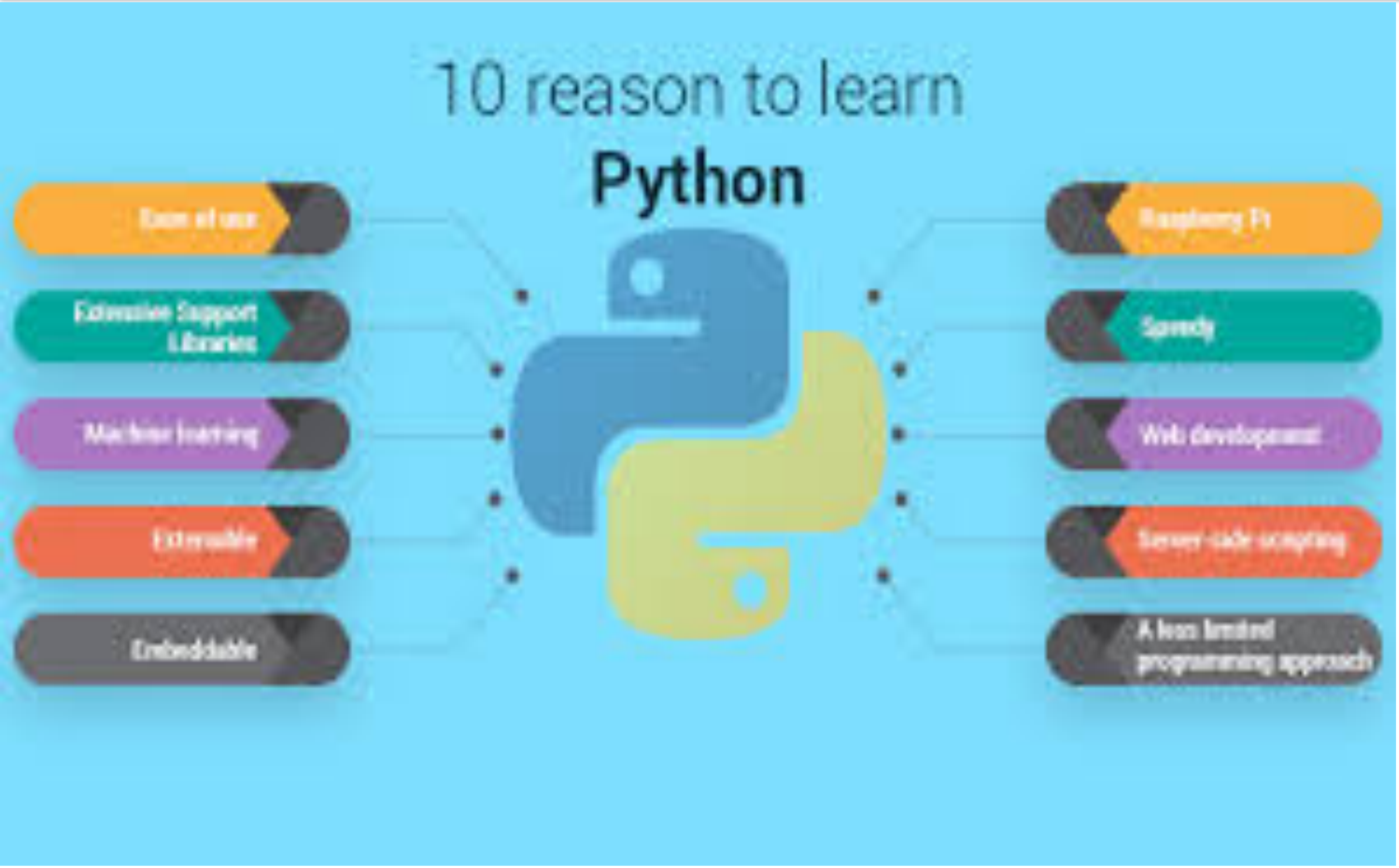
- The style of coding is easy
- The Python is free and open source. No need to pay any subscription charges.
- Python is effectively platform independent
- It is a full programing language
- Python is on the cutting edge, and expanding rapidly, 100,000+ modules are available

Why to use Python

- Python has unrivaled help resources. The community support is overwhelming. There are numerous forums to help you out.
- Python makes the best graphics
- One of the highly sought skill by analytics and data science companies
- comes standard with some of the most flexible and powerful graphics routines available anywhere

There are may more benefits. But , these are the ones which seems to be important and have kept me going.

Cont’d


10 reason to learn
Python
Ease of use
Extension Support Libraries
Machine learning
Extensible
Embeddable
Raspberry Pi
Speedy
Web development
Server-side scripting
A less limited programming approach

Cont’d

	Development **Backend – API – FrontEnd**	**Big Data** **Transformation – AI – ML**	**Networking** **Scripting – Security – Automation**
Job Types	Python Developer Software Developer Full-Stack Developer	Data Scientist Data Analyst Research Analyst	DevOps Engineer
Popular Libraries	**Libraries** Requests – Scrapy – Pillow Nose **Web Frameworks** Django – Flask – Tornado Falcon – Hug	**Transformation - Statistics** Pandas – NumPy **Visualization** Seaborn – Plotly **Machine Learning - AI** Scikit-Learn – Tensor Flow NLTK	Ansible Netmiko NAPALM Pyeapi Junos PyEZ Twisted Scapy
IDE	PyDev (Eclipse) – PyCharm – VIM Spyder Python – Komodo IDE – PTVS Eric Python – Sublime – Emacs		

How to install Python?

There are two approaches to install Python?

- You can download Python directly from its project site “python.org/downloads/” and install individual components and libraries you want.

- Alternately, you can download and install a package, which comes with pre-installed libraries (Anaconda, anaconda.com/distribution). This method provides hassle free installation.

Choosing a development environment

Once you have installed Python, there are various options for choosing an environment. Following are the most common options.

- Terminal / Shell based
- IDLE
- iPython/Jupyter notebook

Running the Jupyter Notebook

One of the major components of the Jupyter project is the notebook, a type of interactive document for code, text, data visualizations, and other output. The Jupyter notebook interacts with *kernels*, which are implementations of the Jupyter interactive computing protocol in any number of programming languages.

To start up Jupyter, run the command jupyter notebook in a terminal

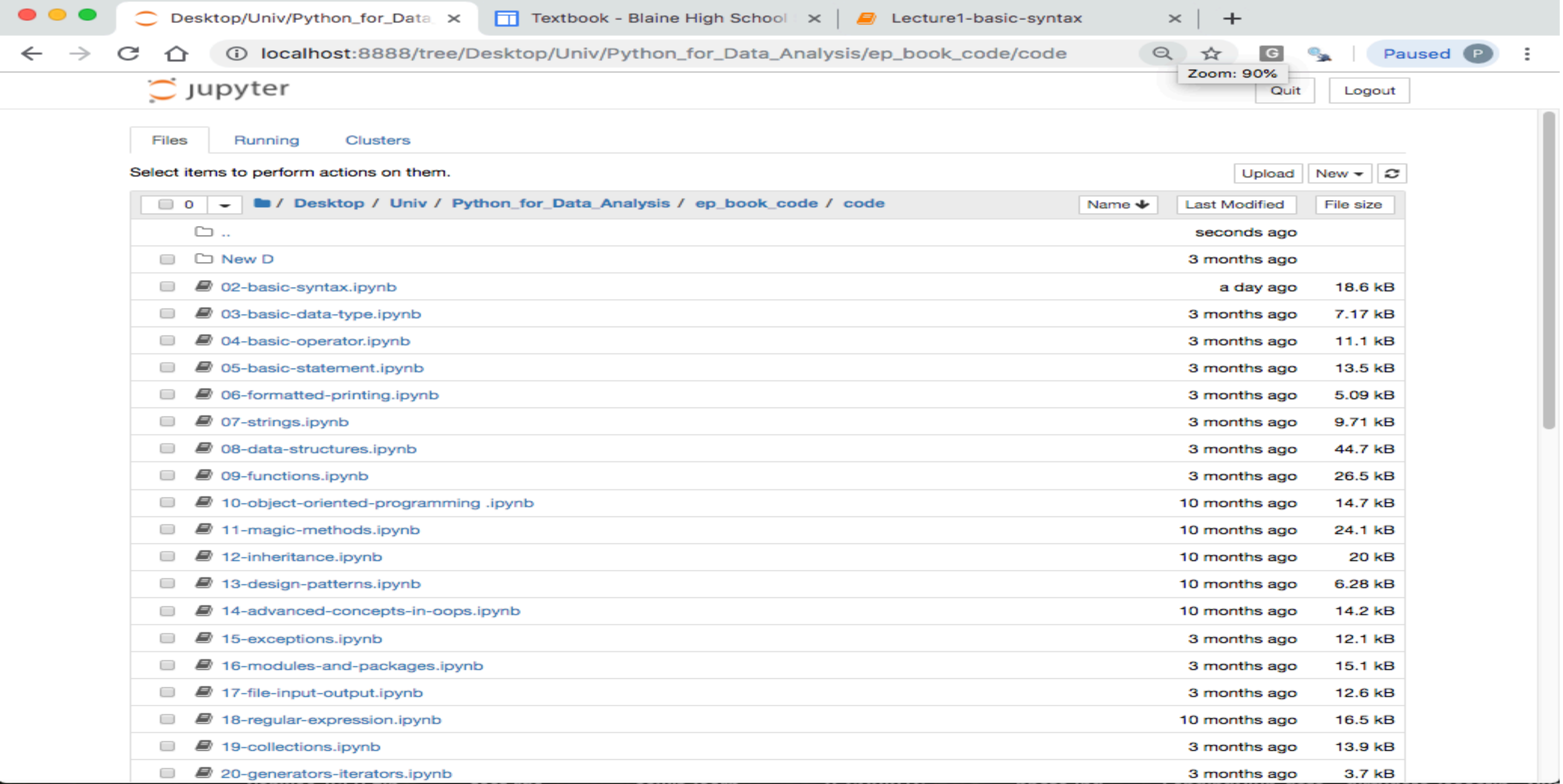
```
 $ jupyter notebook
```

```
homes-MBP:~ home$ jupyter notebook
/Users/home/anaconda3/lib/python3.6/site-packages/notebook/
services/kernels/kernelmanager.py:19: VisibleDeprecationWarning:
zmq.eventloop.minitornado is deprecated in pyzmq 14.0 and will be
removed.
    Install tornado itself to use zmq with the tornado IOLoop.

  from jupyter_client.session import Session
[11:30:32.666 NotebookApp]
```

Running the Jupyter Notebook

On many platforms, jupyter will automatically open up in your web browser. Otherwise, you can navigate to the http address printed when you started the notebook.



localhost:8888/tree/Desktop/Univ/Python_for_Data_Analysis/ep_book_code/code

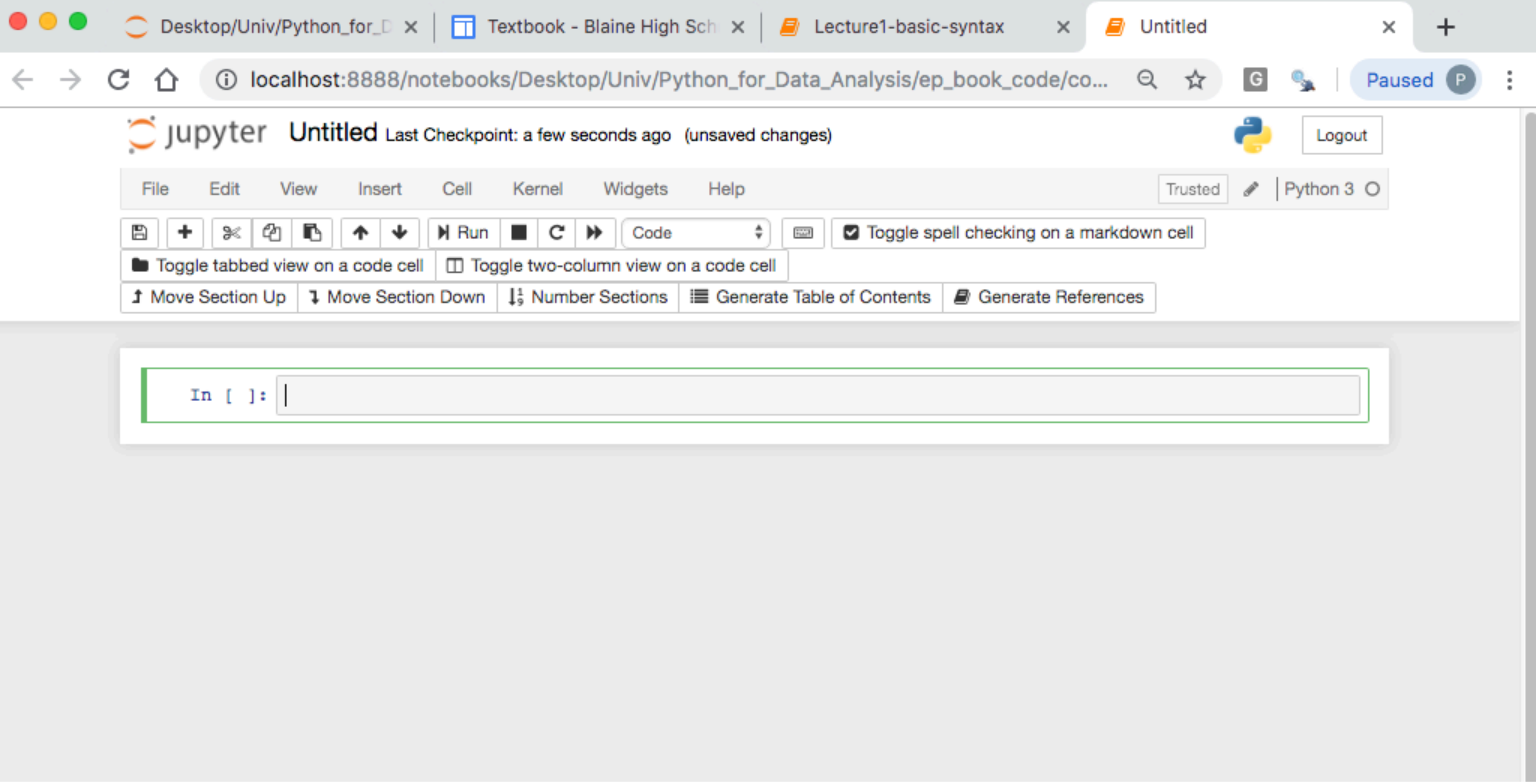
Select items to perform actions on them.

/ Desktop / Univ / Python_for_Data_Analysis / ep_book_code / code

Name	Last Modified	File size
..	seconds ago	
New D	3 months ago	
02-basic-syntax.ipynb	a day ago	18.6 kB
03-basic-data-type.ipynb	3 months ago	7.17 kB
04-basic-operator.ipynb	3 months ago	11.1 kB
05-basic-statement.ipynb	3 months ago	13.5 kB
06-formatted-printing.ipynb	3 months ago	5.09 kB
07-strings.ipynb	3 months ago	9.71 kB
08-data-structures.ipynb	3 months ago	44.7 kB
09-functions.ipynb	3 months ago	26.5 kB
10-object-oriented-programming .ipynb	10 months ago	14.7 kB
11-magic-methods.ipynb	10 months ago	24.1 kB
12-inheritance.ipynb	10 months ago	20 kB
13-design-patterns.ipynb	10 months ago	6.28 kB
14-advanced-concepts-in-oops.ipynb	10 months ago	14.2 kB
15-exceptions.ipynb	3 months ago	12.1 kB
16-modules-and-packages.ipynb	3 months ago	15.1 kB
17-file-input-output.ipynb	3 months ago	12.6 kB
18-regular-expression.ipynb	10 months ago	16.5 kB
19-collections.ipynb	3 months ago	13.9 kB
20-generators-iterators.ipynb	3 months ago	3.7 kB

Running the Jupyter Notebook

To create a new notebook, click the New button and select the “Python 3” option. You should see something like


Desktop/Univ/Python_for_D
Textbook - Blaine High Sch
Lecture1-basic-syntax
Untitled
localhost:8888/notebooks/Desktop/Univ/Python_for_Data_Analysis/ep_book_code/co...
Paused
jupyter
Untitled Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Toggle spell checking on a markdown cell
Toggle tabbed view on a code cell
Toggle two-column view on a code cell
Move Section Up
Move Section Down
Number Sections
Generate Table of Contents
Generate References
In [ ]:

Essential Python Libraries

Following are a list of libraries, you will need for any scientific computations and data analysis:

NumPy short for Numerical Python has long been a cornerstone of numerical computing in Python.

- The most powerful feature of NumPy is n-dimensional array.
- This library also contains linear algebra functions, Fourier transforms, advanced random number capabilities
- Functions for performing element-wise computations with arrays or mathematical operations between arrays
- Tools for reading and writing array-based datasets to disk
- and tools for integration with other low level languages like Fortran, C and C++. Libraries written in a lower-level language can operate on the data stored in a NumPy array without copying data into some other memory representation.

Essential Python Libraries

SciPy stands for Scientific Python. SciPy is built on NumPy.
It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
Latex commands to add math to your plot.

Matplotlib is the most popular Python library for producing plots and other two-dimensional data visualization. It is designed for creating plots suitable for publication.
You can use Pylab feature in ipython notebook (ipython notebook – pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.

Python Libraries

pandas provides high-level data structures and functions designed to make working with structured or tabular data fast, easy, and expressive. Since its emergence in 2010, it is extensively used for data munging and preparation.

Pandas have been instrumental in boosting Python’s usage in data scientist community.

The primary objects in pandas that we will use are the ***DataFrame*** and ***Series.***

Pandas blends the high-performance, array-computing ideas of NumPy with flexible data manipulation capabilities of spreadsheets and relational databases.

It provides sophisticated indexing functionality to make it easy to reshape, slice and dice, perform aggregations, and select subsets of data.

Python Libraries

Scikit-learn for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction, model selection, Preprocessing. Along with pandas, statsmodels, and Ipython, scikit-learn has been critical for enabling Python to be a productive data science programming language.
Statsmodels for statistical modeling. Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.
Compared with scikit-learn, statsmodels contains algorithms for classical (primarily frequent) statistics and econometrics. Statmodels is more focused on statistical inference, providing uncertainty estimates and p-values for parameters, scikit-learn, by contrast, is more prediction

Python Libraries

Seaborn for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.

Python Libraries

You may need some other libraries also depending upon the task:

Bokeh for creating interactive plots, dashboards and data applications on modern web-browsers..

Blaze for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.

Scrapy for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information.

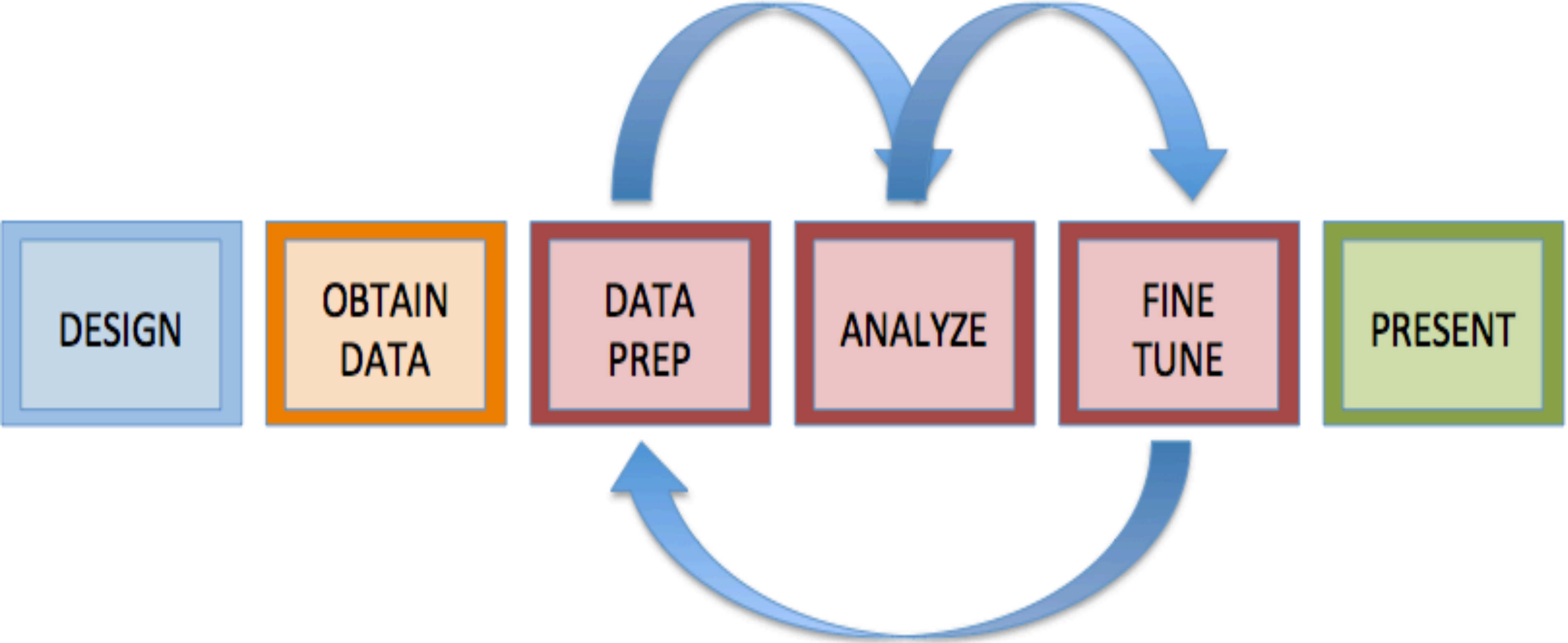
SymPy for symbolic computation. It has wide-ranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. **Requests** for accessing the web. It works similar to the the standard python library urllib2 but is much easier to code.

BeautifulSoup for scrapping web. It is not as good as Scrapy as it extract information from just a single webpage in a run.

Regular expressions for finding patterns in text data

Practical Aspects of Data Analysis

Steps in an analytics exercise



DESIGN

OBTAIN DATA

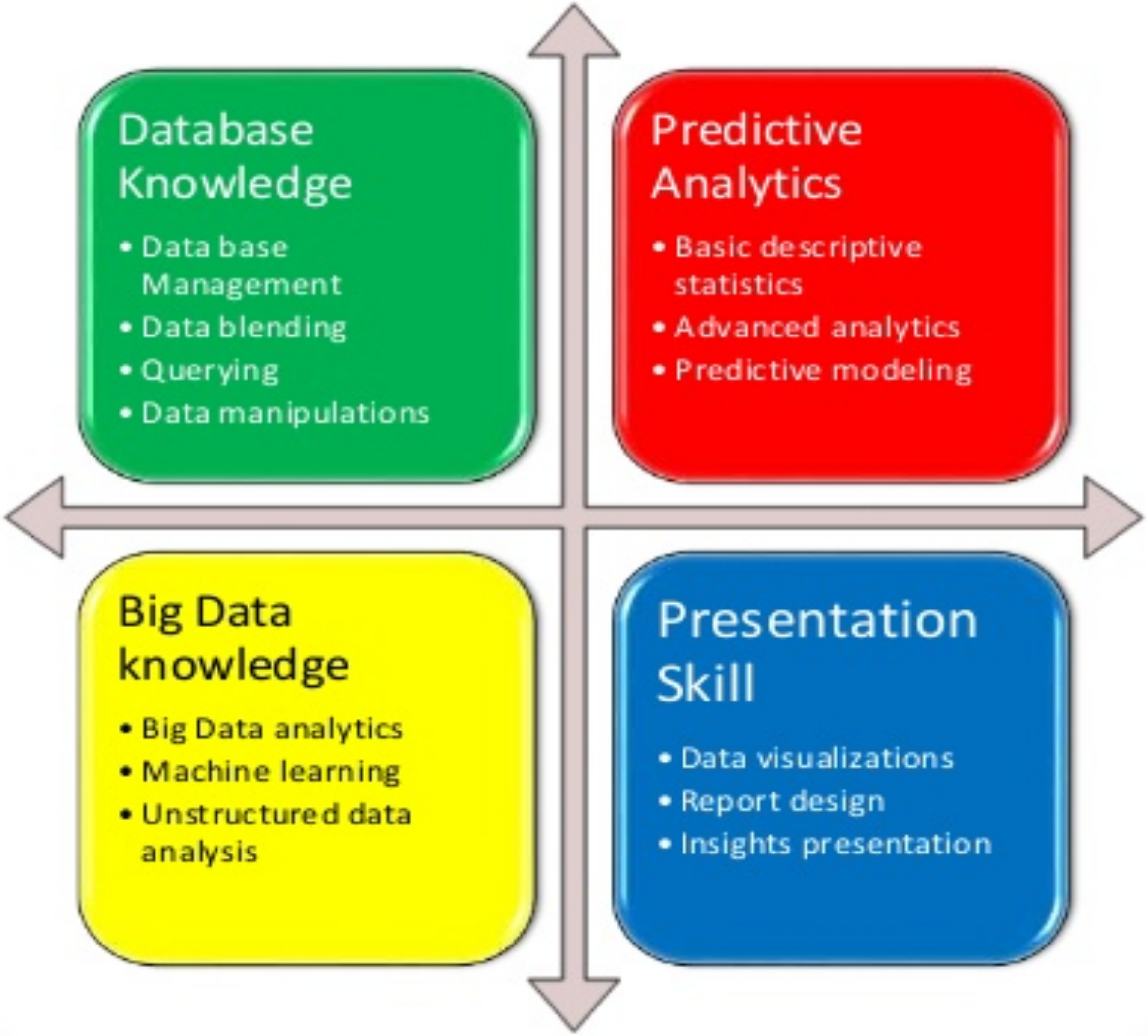
DATA PREP

ANALYZE

FINE TUNE

PRESENT

The Techniques



Database Knowledge

- Data base Management
- Data blending
- Querying
- Data manipulations

Predictive Analytics

- Basic descriptive statistics
- Advanced analytics
- Predictive modeling

Big Data knowledge

- Big Data analytics
- Machine learning
- Unstructured data analysis

Presentation Skill

- Data visualizations
- Report design
- Insights presentation

Venkat Reddy Konasani
http://www.slideshare.net/21_venkat

Analytics philosophy

- You have to get your hands dirty
- Keep trying out things
- Download data, or some code, and try to run
- Make small tweaks
- Analysis is both a science and art.
- Understand how the analysis has been put together
- There is no way to know everything. Learning is the answer
  - You learn by observing and practicing

Starting with the end in mind

- Ask yourself these before you start the analysis
- What do I want to present?
- Which graphs will I create? and how many?
- What data will I need
- Where I can the data, i.e., source of the data
- Try a “mock plot” with dummy data
- Does it look like what I want

Getting comfortable with big data

- Recommend that you work with at least one data set that has >100K rows

- Over the course, you must download and use at least two OpenGov type datasets (To get into the habit: data.gov)

- For either your final project, or any open homework/assignment problem

Resources to Learn Python

- **Python for Data Analysis: Wes McKinney**
- **Python Data Science Handbook: Jake VanderPlas**
- **docs.pytjon.org/3/tutorial**
- **https://learnpython.org**